

DRAFT/PROPOSED/UNOFFICIAL/NOT FOR ENTRYCLAIM LISTING

1. (Currently amended) A method in a data processing system including a computer, comprising the steps of:

providing the computer with a software development tool having a user interface that is operable by a user to automatically reflect a modification in the source code to avoid completely regenerating the source code, wherein the software development tool includes computer instructions for performing the following steps:

receiving an identification of a data structure with an attribute field in a database of data structures useable to form an object-oriented element from the data structure;

determining via the computer ~~and without user interaction~~ whether the data structure is associated with the source code;

~~when it is determined a determination is made~~ that the data structure is associated with the source code, another determin[[ing]]ation is made via the computer ~~as to and without user interaction~~ whether the attribute field of the data structure is associated with an attribute in the source code;[[ and]]

~~when it is determined an alternative determination is made~~ that the attribute field is not associated with an attribute in the source code, generating via the computer ~~and without user interaction~~ a new attribute in the source code from the attribute field;

receiving user input to modify the source code;[[ and]]

modifying and displaying via the computer ~~and without user interaction~~ a graphical representation of at least a portion of modified[[the]] source code to reflect the source code modification, wherein the modified source code and the graphical representation are displayed simultaneously; and

using a QA module to evaluate the modified source code, wherein an error message is generated and displayed if the modified source code does not conform to a predefined or user-defined style.

2. (Currently amended) The method of claim 1, further comprising having the computer perform the following steps ~~without user interaction~~:

BOR-006  
09/839,526DRAFT/PROPOSED/UNOFFICIAL/NOT FOR ENTRY

when it is determined that the data structure is not associated with source code, retrieving a portion of the data structure; and generating the source code from the portion of the data structure.

3. (Currently amended) The method of claim 1, further comprising having the computer perform the following steps ~~without user interaction~~:

when it is determined that the data structure is associated with source code, determining whether a second attribute in the source code is associated with a second attribute field in the data structure; and

when it is determined that a second attribute in the source code is not associated with a second attribute field in the data structure, removing the second attribute from the source code.

4. (Currently amended) The method of claim 3, wherein the step of removing the second attribute from the source code comprises the step of removing via the computer and ~~without user interaction~~ a method associated with the second attribute from the source code.

5. - 6. (Cancelled)

7. (Currently amended) The method of claim 4, further comprising the step of modifying via the computer ~~and without user interaction~~ the graphical representation of the source code to reflect the removal of the second attribute.

8. (Currently amended) The method of claim 1, wherein the step of determining whether the data structure is associated with the source code comprises the step of searching via the computer ~~and without user interaction~~ a comment in the source code for the identification of the data structure.

9. (Currently amended) The method of claim 1, wherein the step of determining whether the data structure is associated with the source code comprises the step of comparing via the computer ~~and without user interaction~~ a name for the source code with the identification of the data structure.

10. (Currently amended) The method of claim 1, further comprising having the computer perform the following steps ~~without user interaction~~:

DRAFT/PROPOSED/UNOFFICIAL/NOT FOR ENTRY

DRAFT/PROPOSED/UNOFFICIAL/NOT FOR ENTRY

retrieving access information for the database; and retrieving a portion of the data structure from the database using the access information.

11. (Currently amended) The method of claim 10, wherein the step of retrieving the access information comprises the step of retrieving via the computer ~~and without user interaction~~ the identification of the data structure and the access information from a configuration file;

12. (Currently amended) The method of claim 10, wherein the step of retrieving the access information comprises the step of retrieving via the computer ~~and without user interaction~~ the identification of the data structure and the access information from a comment of the source code.

13. (Original) The method of claim 10, wherein the portion of the data structure comprises the attribute field of the data structure.

14. (Original) The method of claim 1, wherein the source code comprises a class.

15. (Original) The method of claim 1, wherein the source code comprises a distributed computing component.

16. (Original) The method of claim 15, wherein the distributed computing component is an Enterprise JavaBean.[[TM.]]

17. (Currently amended) The method of claim 1, wherein the step of generating the new attribute in the source code comprises the step of generating via the computer ~~and without user interaction~~ a method in the source code to access the attribute field of the data structure.

18. (Currently amended) A method in a data processing system including a computer having source code that corresponds to a data structure within a database of data structures useable to form an object-oriented element from the data structure on a secondary storage device, the method comprising the steps of:

providing a software development tool having a user interface that is operable by a user to automatically reflect a modification in the source code to avoid completely regenerating the source code, wherein the software development tool includes computer instructions for performing the following steps:

BOR-006  
09/839,526DRAFT/PROPOSED/UNOFFICIAL/NOT FOR ENTRY

receiving an indication to update the source code; determining via the computer ~~and without user interaction~~-whether a first attribute in the source code is associated with a first attribute field in the data structure;

when it is determined that a first attribute in the source code is not associated with a first attribute field in the data structure, removing the first attribute from the source code;

determining via the computer ~~and without user interaction~~-whether a second attribute field in the data structure is associated with a second attribute in the source code; and when it is determined that a second attribute field is not associated with a second attribute in the source code, generating via the computer ~~and without user interaction~~ the second attribute in the source code from the second attribute field and associating the second attribute with the second attribute field;[[ and]]

receiving user input to modify the source code;[[ and]]

modifying and displaying via the computer ~~and without user interaction~~ a graphical representation of at least a portion of modified[[the]] source code to reflect the source code modification, wherein the modified source code and the graphical representation are displayed simultaneously; and

using a QA module to evaluate the modified source code, wherein an error message is generated and displayed if the modified source code does not conform to a predefined or user-defined style.

19. (Currently amended) The method of claim 18, further comprising the step of retrieving via the computer ~~and without user interaction~~-an identification of the data structure and access information for the secondary storage from a comment in the source code.

20. (Currently amended) The method of claim 19, further comprising the step of retrieving via the computer ~~and without user interaction~~-a portion of the data structure from the secondary storage device using the access information.

21. (Original) The method of claim 20, wherein the portion comprises the first and the second attribute fields.

22. (Currently amended) The method of claim 18, wherein the step of removing the first attribute from the source code comprises the step of removing via the computer ~~and without~~

DRAFT/PROPOSED/UNOFFICIAL/NOT FOR ENTRY

BOR-006  
09/839,526DRAFT/PROPOSED/UNOFFICIAL/NOT FOR ENTRY

~~user interaction~~ a method associated with the first attribute from the source code.

23. (Currently amended) The method of claim 18, wherein the step of generating the second attribute in the source code comprises the step of generating via the computer ~~and without~~ ~~user interaction~~ a method in the source code to access the second attribute field of the data structure.

24. (Currently amended) A method in a data processing system including a computer having a memory device with source code and a secondary storage device with a data structure within a database of data structures useable to form an object-oriented element from the data structure corresponding to the source code, the method comprising the steps of: providing a software development tool having a user interface that is operable by a user to automatically reflect a modification in the source code to avoid completely regenerating the source code, wherein the software development tool includes computer instructions for performing the following steps:

receiving an indication that the data structure has been modified; and automatically reflecting the modification in the source code via the computer ~~and without user interaction~~ so as to avoid completely regenerating the source code;

receiving user input to modify the source code;[[ and]]

modifying and displaying via the computer ~~and without user interaction~~ a graphical representation of at least a portion of modified[[the]] source code to reflect the source code modification, wherein the modified source code and the graphical representation are displayed simultaneously; and

using a QA module to evaluate the modified source code, wherein an error message is generated and displayed if the modified source code does not conform to a predefined or user-defined style.

25. (Currently amended) The method of claim 24, wherein the step of automatically reflecting the modification comprises having the computer perform the following steps ~~without~~ ~~user interaction~~:

determining whether a first attribute in the source code is associated with a first attribute field in the data structure; and when it is determined that a first attribute in the source code is not

DRAFT/PROPOSED/UNOFFICIAL/NOT FOR ENTRY

DRAFT/PROPOSED/UNOFFICIAL/NOT FOR ENTRY

associated with a first attribute field in the data structure, removing the first attribute from the source code.

26. (Currently amended) The method of claim 25, wherein the step of removing the first attribute from the source code comprises the step of removing via the computer ~~and without user interaction~~ a first method associated with the first attribute in the source code.

27. (Currently amended) The method of claim 25, wherein the step of automatically reflecting the modification further comprises having the computer perform the following steps ~~without user interaction~~:

determining whether a second attribute field in the data structure is associated with a second attribute in the source code; and

when it is determined that a second attribute field in the data structure is not associated with a second attribute in the source code, generating the second attribute in the source code from the second attribute field and associating the second attribute with the second attribute field.

28. (Currently amended) The method of claim 27, wherein the step of generating the second attribute in the source code comprises the step of generating via the computer ~~and without user interaction~~ a second method in the source code to access the second attribute field of the data structure.

29 - 30. (Cancelled)

31. (Original) The method of claim 24, wherein the source code comprises a class.

32. (Original) The method of claim 24, wherein the source code comprises a distributed computing component.

33. (Original) The method of claim 32, wherein the distributed computing component is an Enterprise JavaBean.([TM..])

34. (Currently amended) A method in a data processing system including a computer having a memory device with source code, the method comprising the steps of:

providing a software development tool having a user interface that is operable by a user to automatically reflect a modification in the source code to avoid completely regenerating the source code, wherein the software development tool includes computer instructions for performing the following steps:

BOR-006  
09/839,526DRAFT/PROPOSED/UNOFFICIAL/NOT FOR ENTRY

determining via the computer ~~and without user interaction~~ whether the source code is associated with a data structure within a database of data structures useable to form an object-oriented element from the data structure;

when it is determined that the source code is associated with the data structure,

determining via the computer ~~and without user interaction~~ whether a first attribute in the source code is associated with a first attribute field of the data structure;

when it is determined that the first attribute in the source code is not associated with the first attribute field in the data structure, generating via the computer ~~and without user interaction~~ the first attribute field in the data structure;

determining via the computer ~~and without user interaction~~ whether a second attribute field in the data structure is associated with a second attribute in the source code;[[ and]]

when it is determined that the second attribute field is not associated with the second attribute in the source code, removing via the computer ~~and without user interaction~~ the second attribute field from the data structure;[[ and]]

receiving user input to modify the source code;[[ and]]

modifying and displaying via the computer ~~and without user interaction~~ a graphical representation of at least a portion of modified[[the]] source code to reflect the source code modification, wherein the modified source code and the graphical representation are displayed simultaneously; and

using a QA module to evaluate the modified source code, wherein an error message is generated and displayed if the modified source code does not conform to a predefined or user-defined style.

35. (Currently amended) The method of claim 34, further comprising the step of when it is determined that the source code is not associated with the data structure, generating via the computer ~~and without user interaction~~ the data structure from the source code.

36. (Currently amended) The method of claim 34, wherein the step of determining whether the source code is associated with the data structure comprises the step of searching via the computer ~~and without user interaction~~ a comment in the source code for an identification of the data structure.

DRAFT/PROPOSED/UNOFFICIAL/NOT FOR ENTRY

DRAFT/PROPOSED/UNOFFICIAL/NOT FOR ENTRY

37. (Currently amended) The method of claim 34, wherein the step of determining whether the source code is associated with the data structure comprises the step of comparing via the computer ~~and without user interaction~~ a name for the source code with an identification of the data structure.

38. (Currently amended) The method of claim 34, further comprising having the computer perform the following steps ~~without user interaction~~:

retrieving access information for a database that stores the data structure; and retrieving a portion of the data structure from the database using the access information.

39. (Currently amended) The method of claim 38, wherein the step of retrieving the access information comprises the step of retrieving via the computer ~~and without user interaction~~ an identification of the data structure and the access information from a configuration file.

40. (Currently amended) The method of claim 38, wherein the step of retrieving the access information comprises the step of retrieving via the computer ~~and without user interaction~~ an identification of the data structure and the access information from a comment of the source code.

41. (Original) The method of claim 38, wherein the portion of the data structure comprises the first attribute field in the data structure.

42. (Original) The method of claim 34, wherein the source code comprises a class.

43. (Original) The method of claim 34, wherein the source code comprises a distributed computing component.

44. (Original) The method of claim 34, wherein the first attribute field in the data structure is related to a method in the source code.

45. (Currently amended) A method in a data processing system having a memory device with source code, the method comprising the steps of:

providing a software development tool having a user interface that is operable by a user to automatically modify source code, wherein the software development tool includes computer instructions for performing the following steps[[.]]:

receiving an indication to update a data structure within a database of data structures useable to form an object-oriented element from the data structure related to the source code;

DRAFT/PROPOSED/UNOFFICIAL/NOT FOR ENTRY

determining via the computer and without user interaction whether a first attribute field of the data structure is associated with a first attribute in the source code;

when it is determined that the first attribute field of the data structure is not associated with the first attribute of the source code, removing the first attribute field from the data structure;

determining via the computer and without user interaction whether a second attribute in the source code is associated with a second attribute field in the data structure;[[ and]]

when it is determined that the second attribute is not associated with the second attribute field in the data structure, adding the second attribute field to the data structure; and receiving user input to modify at least a portion of modified[[the]] source code;[[ and]]

modifying and displaying via the computer and without user interaction a graphical representation of the source code to reflect the source code modification, wherein the modified source code and the graphical representation are displayed simultaneously; and

using a QA module to evaluate the modified source code, wherein an error message is generated and displayed if the modified source code does not conform to a predefined or user-defined style.

46. (Currently amended) The method of claim 45, further having the computer perform the following steps without user interaction:

retrieving access information for a database that stores the data structure; and retrieving a portion of the data structure from the database using the access information.

47. (Currently amended) The method of claim 46, wherein the step of retrieving the access information comprises the step of retrieving via the computer and without user interaction an identification of the data structure and the access information from a configuration file.

48. (Currently amended) The method of claim 46, wherein the step of retrieving the access information comprises the step of retrieving via the computer and without user interaction an identification of the data structure and the access information from a comment of the source code.

49. (Original) The method of claim 46, wherein the portion of the data structure comprises the first attribute field in the data structure.

BOR-006  
09/839,526DRAFT/PROPOSED/UNOFFICIAL/NOT FOR ENTRY

50. (Original) The method of claim 45, wherein the source code comprises a class.

51. (Original) The method of claim 45, wherein the source code comprises a distributed computing component.

52. (Currently amended) A method in a data processing system having a memory device with source code and a secondary storage device with a data structure within a database of data structures useable to form an object-oriented element from the data structure corresponding to the source code, the method comprising the steps of:  
providing a software development tool having a user interface that is operable by a user to automatically reflect a modification in the source code to avoid completely regenerating the source code, wherein the software development tool includes computer instructions for performing the following steps:  
receiving an indication that the source code has been modified; and automatically reflecting via the computer ~~and without user interaction~~ the modification in the data structure so as to avoid completely regenerating the data structure;  
receive user input to modify the source code;[[ and]]  
modifying and displaying via the computer ~~and without user interaction~~ a graphical representation of at least a portion of modified[[the]] source code to reflect the source code modification, wherein the modified source code and the graphical representation are displayed simultaneously; and  
using a QA module to evaluate the modified source code, wherein an error message is generated and displayed if the modified source code does not conform to a predefined or user-defined style.

53. (Currently amended) The method of claim 52, wherein the step of automatically reflecting the modification comprises having the computer perform the following steps ~~without user interaction~~:  
determining whether a first attribute in the source code is associated with a first attribute field of the data structure; and when it is determined that a first attribute is not associated with a first attribute field in the data structure, generating the first attribute field in the data structure.

54. (Currently amended) The method of claim 52, further comprising having the

DRAFT/PROPOSED/UNOFFICIAL/NOT FOR ENTRY

BOR-006  
09/839,526DRAFT/PROPOSED/UNOFFICIAL/NOT FOR ENTRY

computer) perform the following steps ~~without user interaction~~: determining whether a second attribute field in the data structure is associated with a second attribute in the source code; and when it is determined that a second attribute field is not associated with a second attribute in the source code, removing the second attribute field from the data structure.

55. (Currently amended) The method of claim 52, further having the computer perform the following steps ~~without user interaction~~:

retrieving access information for a database that stores the data structure; and retrieving a portion of the data structure from the database using the access information.

56. (Currently amended) The method of claim 55, wherein the step of retrieving the access information comprises the step of retrieving via the computer ~~and without user interaction~~ an identification of the data structure and the access information from a configuration file.

57. (Currently amended) The method of claim 55, wherein the step of retrieving the access information comprises the step of retrieving via the computer ~~and without user interaction~~ an identification of the data structure and the access information from a comment of the source code.

58. (Original) The method of claim 55, wherein the portion of the data structure comprises the first attribute field in the data structure.

59. (Original) The method of claim 52, wherein the source code comprises a class.

60. (Original) The method of claim 52, wherein the source code comprises a distributed computing component.

61. (Original) The method of claim 52, wherein the first attribute field in the data structure is related to a method in the source code.

62. (Currently amended) A computer-readable medium containing instructions for controlling a data processing system to perform a method, the method comprising the steps of: providing a software development tool having a user interface that is operable by a user to automatically reflect a modification in the source code to avoid completely regenerating the source code, wherein the software development tool includes computer instructions for performing the following steps:

DRAFT/PROPOSED/UNOFFICIAL/NOT FOR ENTRY

DRAFT/PROPOSED/UNOFFICIAL/NOT FOR ENTRY

receiving an identification of a data structure with an attribute field in a database of data structures useable to form and object-oriented element from the data structure; determining via the computer ~~and without user interaction~~ whether the data structure is associated with source code;

when it is determined that the data structure is associated with the source code, determining via the computer ~~and without user interaction~~ whether the attribute field of the data structure is associated with an attribute in the source code;[[ and]]

when it is determined that the attribute field is not associated with an attribute in the source code, generating via the computer ~~and without user interaction~~ a new attribute in the source code from the attribute field; and receiving user input to modify the source code; and

modifying and displaying via the computer ~~and without user interaction~~ a graphical representation of at least a portion of modified[[the]] source code to reflect the source code modification, wherein the modified source code and the graphical representation are displayed simultaneously; and

using a QA module to evaluate the modified source code, wherein an error message is generated and displayed if the modified source code does not conform to a predefined or user-defined style.

63. (Currently amended) The computer-readable medium of claim 62, wherein the method further comprises having the computer perform the following steps ~~without user interaction~~:

when it is determined that the data structure is not associated with source code, retrieving a portion of the data structure; and generating the source code from the portion of the data structure.

64. (Currently amended) The computer-readable medium of claim 62, wherein the method further comprises having the computer perform the following steps ~~without user interaction~~:

when it is determined that the data structure is associated with source code, determining whether a second attribute in the source code is associated with a second attribute field in the data structure; and

DRAFT/PROPOSED/UNOFFICIAL/NOT FOR ENTRY

when it is determined that a second attribute in the source code is not associated with a second attribute field in the data structure, removing the second attribute from the source code.

65. (Currently amended) The computer-readable medium of claim 64, wherein the step of removing the second attribute from the source code comprises the step of removing via the computer ~~and without user interaction~~ a method associated with the second attribute from the source code.

66. – 67. (Cancelled)

68. (Currently amended) The computer-readable medium of claim 65, wherein the method further comprises the step of modifying via the computer ~~and without user interaction~~ the graphical representation of the source code to reflect the removal of the second attribute.

69. (Currently amended) The computer-readable medium of claim 62, wherein the step of determining whether the data structure is associated with the source code comprises the step of searching via the computer ~~and without user interaction~~ a comment in the source code for the identification of the data structure.

70. (Currently amended) The computer-readable medium of claim 62, wherein the step of determining whether the data structure is associated with the source code comprises the step of comparing via the computer ~~and without user interaction~~ a name for the source code with the identification of the data structure.

71. (Currently amended) The computer-readable medium of claim 62, wherein the method further comprises having the computer perform the following steps ~~without user interaction~~: retrieving access information for the database; and retrieving a portion of the data structure from the database using the access information.

72. (Currently amended) The computer-readable medium of claim 71, wherein the step of retrieving the access information comprises the step of retrieving via the computer ~~and without user interaction~~ the identification of the data structure and the access information from a configuration file.

73. (Currently amended) The computer-readable medium of claim 71, wherein the step of retrieving the access information comprises the step of retrieving via the computer ~~and without user interaction~~ the identification of the data structure and the access information from a

DRAFT/PROPOSED/UNOFFICIAL/NOT FOR ENTRY

comment of the source code.

74. (Original) The computer-readable medium of claim 71, wherein the portion of the data structure comprises the attribute field of the data structure.

75. (Original) The computer-readable medium of claim 62, wherein the source code comprises a class.

76. (Original) The computer-readable medium of claim 62, wherein the source code comprises a distributed computing component.

77. (Original) The computer-readable medium of claim 76, wherein the distributed computing component is an Enterprise JavaBean. [[TM..]]

78. (Currently amended) The computer-readable medium of claim 62, wherein the step of generating the new attribute in the source code comprises the step of generating via the computer ~~and without user interaction~~ a method in the source code to access the attribute field of the data structure.

79. (Currently amended) A computer-readable medium containing instructions for controlling a data processing system to perform a method, the data processing system having source code that corresponds to a data structure within a database of data structures useable to form an object-oriented element from the data structure on a secondary storage device, the method comprising the steps of: providing a software development tool having a user interface that is operable by a user to automatically reflect a modification in the source code to avoid completely regenerating the source code, wherein the software development tool is for performing the following steps:

receiving an indication to update the source code; determining via the computer ~~and without user interaction~~ whether a first attribute in the source code is associated with a first attribute field in the data structure, when it is determined that a first attribute in the source code is not associated with a first attribute field in the data structure, removing via the computer ~~and without user interaction~~ the first attribute from the source code; determining via the computer ~~and without user interaction~~ whether a second attribute field in the data structure is associated with a second attribute in the source code; [[ and]]

BOR-006  
09/839,526DRAFT/PROPOSED/UNOFFICIAL/NOT FOR ENTRY

when it is determined that a second attribute field is not associated with a second attribute in the source code, generating via the computer ~~and without user interaction~~ the second attribute in the source code from the second attribute field and associating via the computer ~~and without user interaction~~ the second attribute with the second attribute field; and receiving user input to automatically reflect a modification in the source code;[[ and]]

modifying and displaying via the computer ~~and without user interaction~~ a graphical representation of at least a portion of modified[[the]] source code to reflect the source code modification, wherein the modified source code and the graphical representation are displayed simultaneously; and

using a QA module to evaluate the modified source code, wherein an error message is generated and displayed if the modified source code does not conform to a predefined or user-defined style.

80. (Currently amended) The computer-readable medium of claim 79, wherein the method further comprises the step of retrieving via the computer ~~and without user interaction~~ an identification of the data structure and access information for the secondary storage from a comment in the source code.

81. (Currently amended) The computer-readable medium of claim 80, wherein the method further comprises the step of retrieving via the computer ~~and without user interaction~~ a portion of the data structure from the secondary storage device using the access information.

82. (Original) The computer-readable medium of claim 81, wherein the portion comprises the first and the second attribute fields.

83. (Currently amended) The computer-readable medium of claim 79, wherein the step of removing the first attribute from the source code comprises the step of removing via the computer ~~and without user interaction~~ a method associated with the first attribute from the source code.

84. (Currently amended) The computer-readable medium of claim 79, wherein the step of generating the second attribute in the source code comprises the step of generating via the computer ~~and without user interaction~~ a method in the source code to access the second attribute field of the data structure.

DRAFT/PROPOSED/UNOFFICIAL/NOT FOR ENTRY

DRAFT/PROPOSED/UNOFFICIAL/NOT FOR ENTRY

85. (Currently amended) A computer-readable medium containing instructions for controlling a data processing system including a computer to perform a method, the data processing system having a memory device with source code and a secondary storage device with a data structure within a database of data structures useable to form an object-oriented element from the data structure corresponding to the source code, the method comprising the steps of:

providing a software development tool having a user interface that is operable by a user to automatically reflect a modification in the source code to avoid completely regenerating the source code, wherein the software development tool is for performing the following steps: receiving an indication that the data source has been modified;[[ and]]

automatically reflecting via the computer ~~and without user interaction~~ the modification in the source code so as to avoid completely regenerating the source code;

receiving user input to modify the source code;[[ and]]

modifying and displaying via the computer ~~and without user interaction~~ a graphical representation of at least a portion of modified[[the]] source code to reflect the source code modification, wherein the modified source code and the graphical representation are displayed simultaneously; and

using a QA module to evaluate the modified source code, wherein an error message is generated and displayed if the modified source code does not conform to a predefined or user-defined style.

86. (Currently amended) The computer-readable medium of claim 85, wherein the step of automatically reflecting the modification comprises having the computer perform the following steps ~~without user interaction~~:

determining whether a first attribute in the source code is associated with a first attribute field in the data structure; and when it is determined that a first attribute in the source code is not associated with a first attribute field in the data structure, removing the first attribute from the source code.

87. (Currently amended) The computer-readable medium of claim 86, wherein the step of removing the first attribute from the source code comprises the step of removing via the computer ~~and without user interaction~~ a first method associated with the first attribute in the

DRAFT/PROPOSED/UNOFFICIAL/NOT FOR ENTRY

source code.

88. (Currently amended) The computer-readable medium of claim 86, wherein the step of automatically reflecting the modification further comprises having the computer perform the following steps ~~without user interaction~~:

determining whether a second attribute field in the data structure is associated with a second attribute in the source code; and

when it is determined that a second attribute field in the data structure is not associated with a second attribute in the source code, generating the second attribute in the source code from the second attribute field and associating the second attribute with the second attribute field.

89. (Currently amended) The computer-readable medium of claim 88, wherein the step of generating the second attribute in the source code comprises the step of generating via the computer ~~and without user interaction~~ a second method in the source code to access the second attribute field of the data structure.

90.-91. (Cancelled)

92. (Original) The computer-readable medium of claim 85, wherein the source code comprises a class.

93. (Original) The computer-readable medium of claim 85, wherein the source code comprises a distributed computing component.

94. (Original) The computer-readable medium of claim 93, wherein the distributed computing component is an Enterprise JavaBean.([TM..])

95. (Currently amended) A computer-readable medium containing instructions for controlling a data processing system including a computer to perform a method, the data processing system having a memory device with source code, the method comprising the steps of: providing a software development tool having a user interface that is operable by a user to automatically reflect a modification in the source code to avoid completely regenerating the source code, wherein the software development tool includes computer instructions for performing the following steps:

DRAFT/PROPOSED/UNOFFICIAL/NOT FOR ENTRY

determining via the computer ~~and without user interaction~~ whether the source code is associated with a data structure within a database of data structures useable to form an object-oriented element from the data structure;

when it is determined that the source code is associated with the data structure, determining via the computer ~~and without user interaction~~ whether a first attribute in the source code is associated with a first attribute field of the data structure;

when it is determined that the first attribute in the source code is not associated with the first attribute field in the data structure, generating the first attribute field in the data structure; determining via the computer ~~and without user interaction~~ whether a second attribute field in the data structure is associated with a second attribute in the source code;[[ and]]

when it is determined that the second attribute field is not associated with the second attribute in the source code, removing via the computer ~~and without user interaction~~ the second attribute field from the data structure; and receiving user input to modify the source code;[[ and]]

modifying and displaying via the computer ~~and without user interaction~~ a graphical representation of at least a portion of modified[[the]] source code to reflect the source code modification, wherein the modified source code and the graphical representation are displayed simultaneously; and

using a QA module to evaluate the modified source code, wherein an error message is generated and displayed if the modified source code does not conform to a predefined or user-defined style.

96. (Currently amended) The computer-readable medium of claim 95, wherein the method further comprises the step of when it is determined that the source code is not associated with the data structure, generating via the computer ~~and without user interaction~~ the data structure from the source code.

97. (Currently amended) The computer-readable medium of claim 95, wherein the step of determining whether the source code is associated with the data structure comprises the step of searching via the computer ~~and without user interaction~~ a comment in the source code for an identification of the data structure.

98. (Currently amended) The computer-readable medium of claim 95, wherein the

BOR-006  
09/839,526DRAFT/PROPOSED/UNOFFICIAL/NOT FOR ENTRY

step of determining whether the source code is associated with the data structure comprises the step of comparing via the computer and ~~without user interaction~~ a name for the source code with an identification of the data structure.

99. (Currently amended) The computer-readable medium of claim 95, wherein the method further comprises having the computer perform the following steps ~~without user interaction~~:

retrieving access information for a database that stores the data structure; and  
retrieving a portion of the data structure from the database using the access information.

100. (Currently amended) The computer-readable medium of claim 99, wherein the step of retrieving the access information comprises the step of retrieving via the computer and ~~without user interaction~~ an identification of the data structure and the access information from a configuration file.

101. (Currently amended) The computer-readable medium of claim 99, wherein the step of retrieving the access information comprises the step of retrieving via the computer and ~~without user interaction~~ an identification of the data structure and the access information from a comment of the source code.

102. (Original) The computer-readable medium of claim 99, wherein the portion of the data structure comprises the first attribute field in the data structure.

103. (Original) The computer-readable medium of claim 95, wherein the source code comprises a class.

104. (Original) The computer-readable medium of claim 95, wherein the source code comprises a distributed computing component.

105. (Original) The computer-readable medium of claim 95, wherein the first attribute field in the data structure is related to a method in the source code.

106. (Currently amended) A computer-readable medium containing instructions for controlling a data processing system to perform a method, the data processing system having a memory device with source code, the method comprising the steps of:

providing a software development tool having a user interface that is operable by a user to automatically reflect a modification in the source code to avoid completely regenerating the

DRAFT/PROPOSED/UNOFFICIAL/NOT FOR ENTRY

BOR-006  
09/839,526DRAFT/PROPOSED/UNOFFICIAL/NOT FOR ENTRY

source code, wherein the software development tool includes computer instructions for performing the following steps:

receiving an indication to update a data structure within a database of data structures useable to form an object-oriented element from the data structure related to the source code;

determining via the computer ~~and without user interaction~~ whether a first attribute field of the data structure is associated with a first attribute in the source code;

when it is determined that the first attribute field of the data structure is not associated with the first attribute of the source code, removing via the computer ~~and without user interaction~~ the first attribute field from the data structure; determining via the computer ~~and without user interaction~~ whether a second attribute in the source code is associated with a second attribute field in the data structure;[[ and]]

when it is determined that the second attribute is not associated with the second attribute field in the data structure, adding the second attribute field to the data structure; and receiving user input to modify the source code;[[ and]]

modifying and displaying via the computer ~~and without user interaction~~ a graphical representation of at least a portion of modified[[the]] source code to reflect the source code modification, wherein the modified source code and the graphical representation are displayed simultaneously; and

using a QA module to evaluate the modified source code, wherein an error message is generated and displayed if the modified source code does not conform to a predefined or user-defined style.

107. (Currently amended) The computer-readable medium of claim 106, wherein the method further comprises having the computer perform the following steps ~~without user interaction~~:

retrieving access information for a database that stores the data structure; and retrieving a portion of the data structure from the database using the access information.

108. (Currently amended) The computer-readable medium of claim 107, wherein the step of retrieving the access information comprises the step of retrieving via the computer ~~and without user interaction~~ an identification of the data structure and the access information from a

BOR-006  
09/839,526DRAFT/PROPOSED/UNOFFICIAL/NOT FOR ENTRY

configuration file.

109. (Currently amended) The computer-readable medium of claim 107, wherein the step of retrieving the access information comprises the step of retrieving via the computer ~~and without user interaction~~ an identification of the data structure and the access information from a comment of the source code.

110. (Original) The computer-readable medium of claim 107, wherein the portion of the data structure comprises the first attribute field in the data structure.

111. (Original) The computer-readable medium of claim 106, wherein the source code comprises a class.

112. (Original) The computer-readable medium of claim 106, wherein the source code comprises a distributed computing component.

113. (Currently amended) A computer-readable medium containing instructions for controlling a data processing system including a computer to perform a method, the data processing system having a memory device with source code and a secondary storage device with a data structure corresponding to the source code, the method comprising the steps of:

providing a software development tool having a user interface that is operable by a user to automatically reflect a modification in the source code to avoid completely regenerating the source code, wherein the software development tool includes computer instructions for performing the following steps:

receiving an indication that the source code has been modified; and automatically reflecting via computer ~~and without user interaction~~ the modification in the data structure so as to avoid completely regenerating the data structure;[[ and]]

receiving user input to modify the source code;[[ and]]

modifying and displaying via the computer ~~and without user interaction~~ a graphical representation of at least a portion of modified[[the]] source code to reflect the source code modification, wherein the modified source code and the graphical representation are displayed simultaneously; and

using a QA module to evaluate the modified source code, wherein an error message is generated and displayed if the modified source code does not conform to a predefined or user-

BOR-006  
09/839,526DRAFT/PROPOSED/UNOFFICIAL/NOT FOR ENTRYdefined style.

114. (Currently amended) The computer-readable medium of claim 113, wherein the step of automatically reflecting the modification comprises having the computer perform the following steps ~~without user interaction~~:

determining whether a first attribute in the source code is associated with a first attribute field of the data structure; and

when it is determined that a first attribute is not associated with a first attribute field in the data structure, generating the first attribute field in the data structure.

115. (Currently amended) The computer-readable medium of claim 113, wherein the method further comprises having the computer perform the following steps ~~without user interaction~~:

determining whether a second attribute field in the data structure is associated with a second attribute in the source code; and

when it is determined that a second attribute field is not associated with a second attribute in the source code, removing the second attribute field from the data structure.

116. (Currently amended) The computer-readable medium of claim 113, wherein the method further comprises having the computer perform the following steps ~~without user interaction~~:

retrieving access information for a database that stores the data structure; and retrieving a portion of the data structure from the database using the access information.

117. (Currently amended) The computer-readable medium of claim 116, wherein the step of retrieving the access information comprises the step of retrieving via the computer ~~and~~ ~~without user interaction~~ an identification of the data structure and the access information from a configuration file.

118. (Currently amended) The computer-readable medium of claim 116, wherein the step of retrieving the access information comprises the step of retrieving via the computer ~~and~~ ~~without user interaction~~ an identification of the data structure and the access information from a comment of the source code.

119. (Original) The computer-readable medium of claim 116, wherein the portion of

BOR-006  
09/839,526DRAFT/PROPOSED/UNOFFICIAL/NOT FOR ENTRY

the data structure comprises the first attribute field in the data structure.

120. (Original) The computer-readable medium of claim 113, wherein the source code comprises a class.

121. (Original) The computer-readable medium of claim 113, wherein the source code comprises a distributed computing component.

122. (Original) The computer-readable medium of claim 113, wherein the first attribute field in the data structure is related to a method in the source code.

123. (Currently amended) A data processing system comprising:

a software development tool having a user interface that is operable by a user to automatically reflect a modification in the source code to avoid completely regenerating the source code;

a secondary storage device further comprising source code that corresponds to a data structure within a database of data structures useable to form an object-oriented element from the data structure on a secondary storage device;

a computer including a memory device further comprising a program that receives an indication to update the source code, that determines whether a first attribute in the source code is associated with a first attribute field in the data structure[, and]];

when determined that a first attribute in the source code is not associated with a first attribute field in the data structure, the program ~~without user interaction~~ removes the first attribute from the source code, the program ~~without user interaction~~ further determines whether a second attribute field in the data structure is associated with a second attribute in the source code[, and]];

when determined that a second attribute field is not associated with a second attribute in the source code, the computer program ~~without user interaction~~ generates the second attribute in the source code from the second attribute field and associating via computer ~~and without user interaction~~ the second attribute with the second attribute field;[ and]]

receiving user input to modify the source code;

modifying and displaying via the computer ~~and without user interaction~~ a graphical representation of at least a portion of modified[[the]] source code to reflect the source code

DRAFT/PROPOSED/UNOFFICIAL/NOT FOR ENTRY

BOR-006  
09/839,526DRAFT/PROPOSED/UNOFFICIAL/NOT FOR ENTRY

modification, wherein the modified source code and the graphical representation are displayed simultaneously; and

using a QA module to evaluate the modified source code, wherein an error message is generated and displayed if the modified source code does not conform to a predefined or user-defined style.

124. (Currently amended) The data processing system of claim 123, wherein the computer program further retrieves via ~~the computer and~~ without user interaction an identification of the data structure and access information for the secondary storage from a comment in the source code.

125. (Currently amended) The data processing system of claim 124, wherein the computer program further retrieves without user interaction a portion of the data structure from the secondary storage device using the access information.

126. (Original) The data processing system of claim 125, wherein the portion comprises the first and the second attribute fields.

127. (Currently amended) The data processing system of claim 123, wherein when the computer program removes the first attribute from the source code, the computer program removes without user interaction a method associated with the first attribute from the source code.

128. (Currently amended) The data processing system of claim 123, wherein when the computer program generates without user interaction the second attribute in the source code, the computer program generates a method in the source code to access the second attribute field of the data structure.

129. (Currently amended) A data processing system comprising:  
a software development tool having a user interface that is operable by a user to automatically reflect a modification in the source code to avoid completely regenerating the source code; a secondary storage device further comprising source code;

a computer including a memory device further comprising a computer program that receives an indication to update a data structure within a database of data structures useable to form an object-oriented element from the data structure related to the source code, that

DRAFT/PROPOSED/UNOFFICIAL/NOT FOR ENTRY

DRAFT/PROPOSED/UNOFFICIAL/NOT FOR ENTRY

determines whether a first attribute field of the data structure is associated with a first attribute in the source code[[, and]]; when determined that the first attribute field of the data structure is not associated with the first attribute of the source code, the computer program removes ~~without user interaction~~ the first attribute field from the data structure, the computer program further determines whether a second attribute in the source code is associated with a second attribute field in the data structure[[, and]]; when determined that the second attribute is not associated with the second attribute field in the data structure, the computer program adds ~~without user interaction~~ the second attribute field to the data structure;

receiving user input to modify the source code;[[ and]]

modifying and displaying ~~without user interaction~~ a graphical representation of at least a portion of modified[[the]] source code to reflect the source code modification, wherein the modified source code and the graphical representation are displayed simultaneously; and

using a QA module to evaluate the modified source code, wherein an error message is generated and displayed if the modified source code does not conform to a predefined or user-defined style.

130. (Currently amended) The data processing system of claim 129, wherein when the program further retrieves ~~without user interaction~~ access information for a database that stores the data structure, and retrieves a portion of the data structure from the database using the access information.

131. (Currently amended) The data processing system of claim 130, wherein when the computer program retrieves ~~without user interaction~~ the access information, and the computer program retrieves ~~without user interaction~~ an identification of the data structure and the access information from a configuration file.

132. (Currently amended) The data processing system of claim 130, wherein when the computer program retrieves the access information, the computer program retrieves ~~without user interaction~~ an identification of the data structure and the access information from a comment of the source code.

BOR-006  
09/839,526DRAFT/PROPOSED/UNOFFICIAL/NOT FOR ENTRY

133. (Original) The data processing system of claim 130, wherein the portion of the data structure comprises the first attribute field in the data structure.

134. (Original) The data processing system of claim 129, wherein the source code comprises a class.

135. (Original) The data processing system of claim 129, wherein the source code comprises a distributed computing component.

136. (Currently amended) A system including a computer having a memory device with source code and a secondary storage device with a data structure within a database of data structures useable to form an object-oriented element from the data structure corresponding to the source code, the system comprising:

a software development tool having a user interface that is operable by a user to modify source code;

wherein the software development tool includes computer instructions for receiving an indication that the data structure has been modified; and computer instruction for automatically reflecting ~~without user interaction~~ the modification in the source code so as to avoid completely regenerating the source code;[[ and]]

wherein the software development tool includes computer instruction for modifying and displaying ~~without user interaction~~ a graphical representation of at least a portion of modified[[the]] source code to reflect the source code modification, wherein the modified source code and the graphical representation are displayed simultaneously; and

a QA module configured to evaluate the modified source code, wherein an error message is generated and displayed if the modified source code does not conform to a predefined or user-defined style.

DRAFT/PROPOSED/UNOFFICIAL/NOT FOR ENTRY